From the book *Networks, Crowds, and Markets: Reasoning about a Highly Connected World*.
By David Easley and Jon Kleinberg. Cambridge University Press, 2010.
Complete preprint on-line at http://www.cs.cornell.edu/home/kleinber/networks-book/

Chapter 14

Link Analysis and Web Search

14.1 Searching the Web: The Problem of Ranking

When you go to Google and type "Cornell," the first result it shows you is www.cornell.edu, the home page of Cornell University. It's certainly hard to argue with this as a first choice, but how did Google "know" that this was the best answer? Search engines determine how to rank pages using automated methods that look at the Web itself, not some external source of knowledge, so the conclusion is that there must be enough information *intrinsic* to the Web and its structure to figure this out.

Before discussing some of the ideas behind the ranking of pages, let's begin by considering a few of the basic reasons why it's a hard problem. First, search is a hard problem for computers to solve in any setting, not just on the Web. Indeed, the field of *information retrieval* [36, 360] has dealt with this problem for decades before the creation of the Web: automated information retrieval systems starting in the 1960s were designed to search repositories of newspaper articles, scientific papers, patents, legal abstracts, and other document collections in reponse to keyword queries. Information retrieval systems have always had to deal with the problem that keywords are a very limited way to express a complex information need; in addition to the fact that a list of keywords is short and inexpressive, it suffers from the problems of *synonymy* (multiple ways to say the same thing, so that your search for recipes involving scallions fails because the recipe you wanted called them "green onions") and *polysemy* (multiple meanings for the same term, so that your search for information about the animal called a jaguar instead produces results primarily about automobiles, football players, and an operating system for the Apple Macintosh.)

For a long time, up through the 1980s, information retrieval was the province of reference

librarians, patent attorneys, and other people whose jobs consisted of searching collections of documents; such people were trained in how to formulate effective queries, and the documents they were searching tended to be written by professionals, using a controlled style and vocabulary. With the arrival of the Web, where everyone is an author and everyone is a searcher, the problems surrounding information retrieval exploded in scale and complexity.

To begin with, the diversity in authoring styles makes it much harder to rank documents according to a common criterion: on a single topic, one can easily find pages written by experts, novices, children, conspiracy theorists — and not necessarily be able to tell which is which. Once upon a time, the fact that someone had the money and resources to produce a professional-looking, typeset, bound document meant that they were very likely (even if not always) someone who could be taken seriously. Today, anyone can create a Web page with high production values.

There is a correspondingly rich diversity in the set of people issuing queries, and the problem of multiple meanings becomes particularly severe. For example, when someone issues the single-word query "Cornell," a search engine doesn't have very much to go on. Did the searcher want information about the university? The university's hockey team? The Lab of Ornithology run by the university? Cornell College in Iowa? The Nobel-Prize-winning physicist Eric Cornell? The same ranking of search results can't be right for everyone.

These represent problems that were also present in traditional information retrieval systems, just taken to new extremes. But the Web also introduces new kinds of problems. One is the dynamic and constantly-changing nature of Web content. On September 11, 2001, many people ran to Google and typed "World Trade Center." But there was a mismatch between what people thought they could get from Google and what they really got: since Google at the time was built on a model in which it periodically collected Web pages and indexed them, the results were all based on pages that were gathered days or weeks earlier, and so the top results were all descriptive pages about the building itself, not about what had occurred that morning. In response to such events, Google and the other main search engines built specialized "News Search" features, which collect articles more or less continuously from a relatively fixed number of news sources, so as to be able to answer queries about news stories minutes after they appear. Even today, such news search features are only partly integrated into the core parts of the search engine interface, and emerging Web sites such as Twitter continue to fill in the spaces that exist between static content and real-time awareness.

More fundamental still, and at the heart of many of these issues, is the fact that the Web has shifted much of the information retrieval question from a problem of *scarcity* to a problem of *abundance*. The prototypical applications of information retrieval in the pre-Web era had a "needle-in-a-haystack" flavor — for example, an intellectual-property attorney might express the information need, "find me any patents that have dealt with the design

of elevator speed regulators based on fuzzy-logic controllers." Such issues still arise today, but the hard part for most Web searches carried out by the general public is in a sense the opposite: to filter, from among an enormous number of relevant documents, the few that are most important. In other words, a search engine has no problem finding and indexing literally millions of documents that are genuinely relevant to the one-word query "Cornell"; the problem is that the human being performing the search is going to want to look at only a few of these. Which few should the search engine recommend?

An understanding of the network structure of Web pages will be crucial for addressing these questions, as we now discuss.

14.2 Link Analysis using Hubs and Authorities

So we're back to our question from the beginning of the chapter: in response to the one-word query "Cornell," what are the clues that suggest Cornell's home page, www.cornell.edu, is a good answer?

Voting by In-Links. In fact, there is a natural way to address this, provided we start from the right perspective. This perspective is to note that there is not really any way to use features purely internal to the page www.cornell.edu to solve this problem: it does not use the word "Cornell" more frequently or more prominently than thousands of other pages. and so there is nothing on the page itself that makes it stand out. Rather, it stands out because of features on other Web pages: when a page is relevant to the query "Cornell," very often www.cornell.edu is among the pages it links to.

This is the first part of the argument that links are essential to ranking: that we can use them to assess the authority of a page on a topic, through the implicit endorsements that other pages on the topic confer through their links to it. Of course, each individual link may have many possible meanings: it may be off-topic; it may convey criticism rather than endorsement; it may be a paid advertisement. It is hard for search engines to automatically assess the intent of each link. But we hope that in aggregate, if a page receives many links from other relevant pages, then it is receiving a kind of collective endorsement.

In the case of the query "Cornell," we could operationalize this by first collecting a large sample of pages that are relevant to the query — as determined by a classical, text-only, information retrieval approach. We could then let pages in this sample "vote" through their links: which page on the Web receives the greatest number of in-links from pages that are relevant to Cornell? Even this simple measure of link-counting works quite well for queries such as "Cornell," where, ultimately, there is a single page that most people agree should be ranked first.

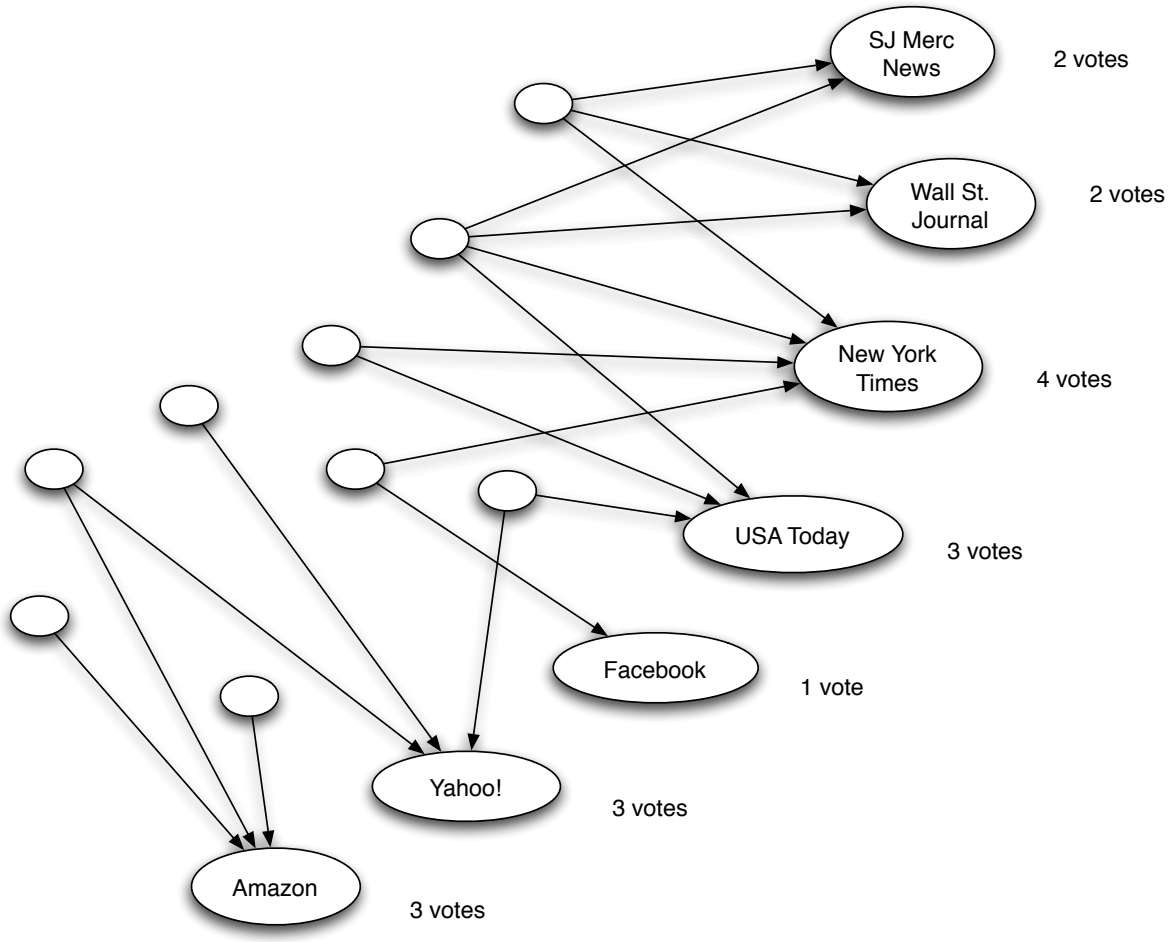

Figure 14.1: Counting in-links to pages for the query "newspapers."

A List-Finding Technique. It's possible to make deeper use of the network structure than just counting in-links, and this brings us to the second part of the argument that links are essential. Consider, as a typical example, the one-word query "newspapers." Unlike the query "Cornell," there is not necessarily a single, intuitively "best" answer here; there are a number of prominent newspapers on the Web, and an ideal answer would consist of a list of the most prominent among them. With the query "Cornell," we discussed collecting a sample of pages relevant to the query and then let them vote using their links. What happens if we try this for the query "newspapers"?

What you will typically observe, if you try this experiment, is that you get high scores for a mix of prominent newspapers (i.e. the results you'd want) along with pages that are going to receive a lot of in-links no matter what the query is — pages like Yahoo!, Facebook, Amazon, and others. In other words, to make up a very simple hyperlink structure for purposes of

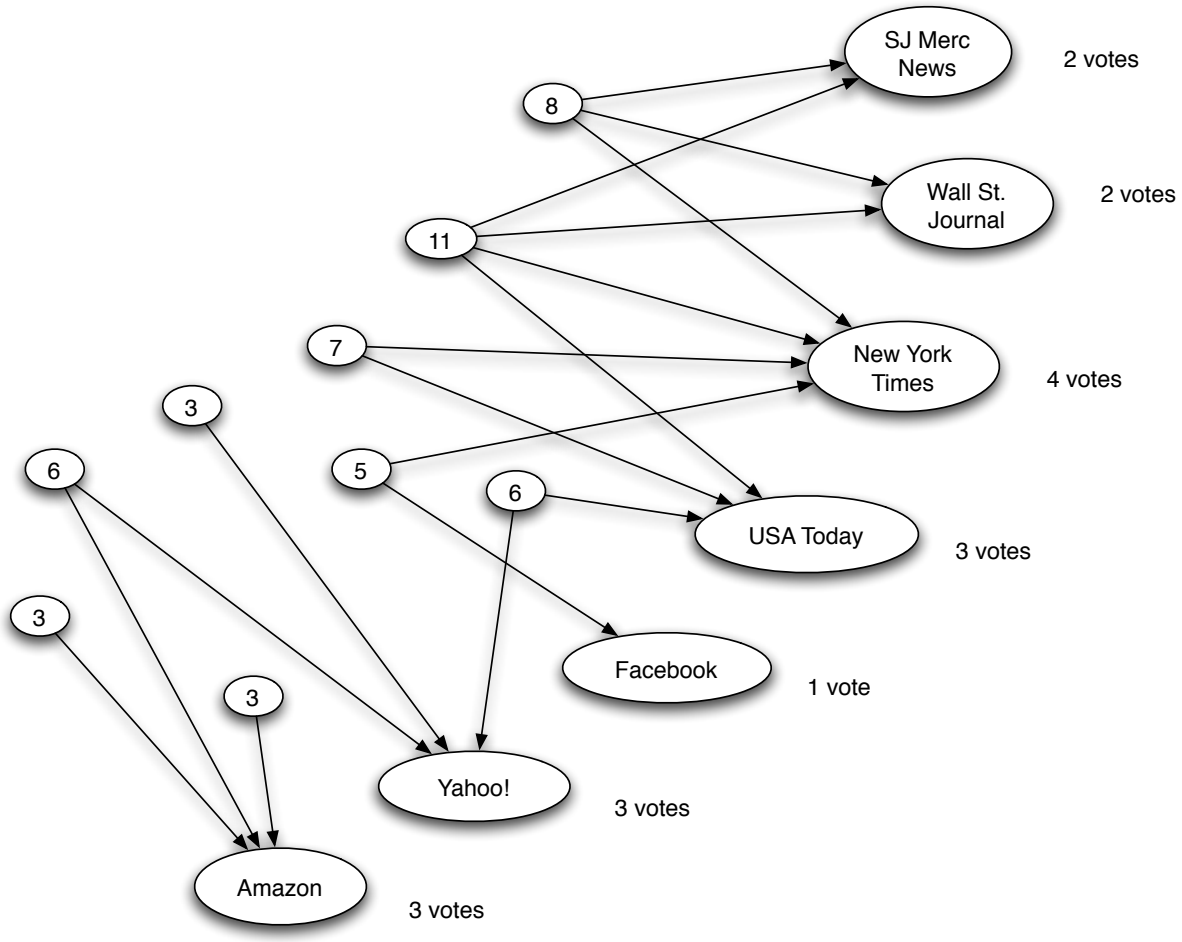

Figure 14.2: Finding good lists for the query "newspapers": each page's value as a list is written as a number inside it.

this example, we'd see something like Figure 14.1: the unlabeled circles represent our sample of pages relevant to the query "newspapers," and among the four pages receiving the most votes from them, two are newspapers (New York Times and USA Today) and two are not (Yahoo! and Amazon). This example is designed to be small enough to try by hand; in a real setting, of course there would be many plausible newspaper pages and many more off-topic pages.

But votes are only a very simple kind of measure that we can get from the link structure — there is much more to be discovered if we look more closely. To try getting more, we ask a different question. In addition to the newspapers themselves, there is another kind of useful answer to our query: pages that compile lists of resources relevant to the topic. Such pages exist for most broad enough queries: for "newspapers," they would correspond to lists

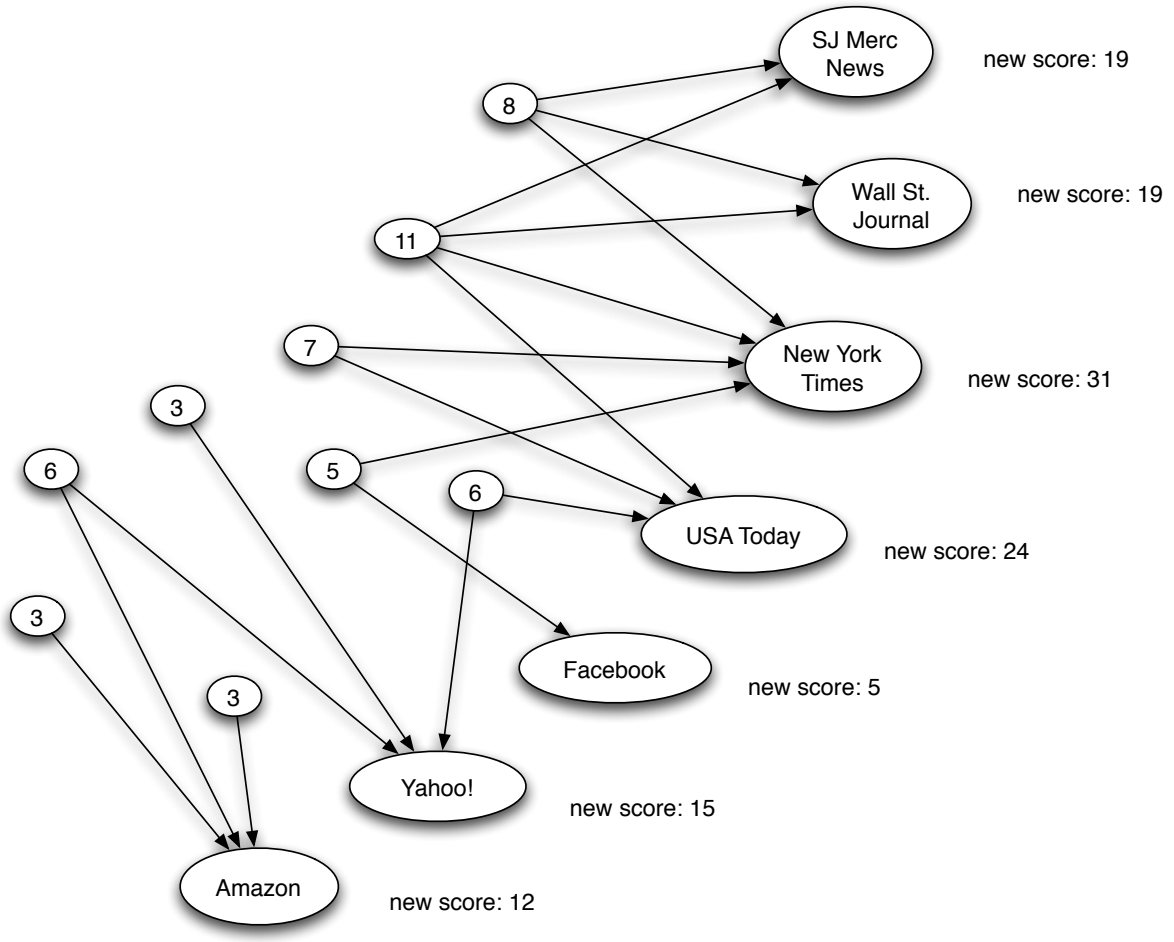

Figure 14.3: Re-weighting votes for the query "newspapers": each of the labeled page's new score is equal to the sum of the values of all lists that point to it.

of links to on-line newspapers; for "Cornell," one can find many alumni who maintain pages with links to the University, its hockey team, its Medical School, its Art Museum, and so forth. If we could find good list pages for newspapers, we would have another approach to the problem of finding the newspapers themselves.

In fact, the example in Figure 14.1 suggests a useful technique for finding good lists. We notice that among the pages casting votes, a few of them in fact voted for *many* of the pages that received a lot of votes. It would be natural, therefore, to suspect that these pages have some sense where the good answers are, and to score them highly as lists. Concretely, we could say that a page's value as a list is equal to the sum of the votes received by all pages that it voted for. Figure 14.2 shows the result of applying this rule to the pages casting votes in our example.

The Principle of Repeated Improvement. If we believe that pages scoring well as lists actually have a better sense for where the good results are, then we should weight their votes more heavily. So, in particular, we could tabulate the votes again, but this time giving each page's vote a weight equal to its value as a list. Figure 14.3 shows what happens when we do this on our example: now the other newspapers have surpassed the initially high-scoring Yahoo! and Amazon, because these other newspapers were endorsed by pages that were estimated to be good lists.

In fact, you can recognize the intuition behind this re-weighting of votes in the way we evaluate endorsements in our everyday lives. Suppose you move to a new town and hear restaurant recommendations from a lot of people. After discovering that certain restaurants get mentioned by a lot of people, you realize that certain *people* in fact had mentioned most of these highly-recommended restaurants when you asked them. These people play the role of the high-value lists on the Web, and it's only natural to go back and take more seriously the more obscure restaurants that they recommended, since you now particularly trust their judgment. This last step is exactly what we are doing in re-weighting the votes for Web pages.

The final part of the argument for link analysis is then the following: Why stop here? If we have better votes on the right-hand-side of the figure, we can use these to get still more refined values for the quality of the lists on the left-hand-side of the figure. And with more refined estimates for the high-value lists, we can re-weight the votes that we apply to the right-hand-side once again. The process can go back and forth forever: it can be viewed as a *Principle of Repeated Improvement*, in which each refinement to one side of the figure enables a further refinement to the other.

Hubs and Authorities. This suggests a ranking procedure that we can try to make precise, as follows [247]. First, we'll call the kinds of pages we were originally seeking — the prominent, highly endorsed answers to the queries — the *authorities* for the query. We'll call the high-value lists the *hubs* for the query. Now, for each page p, we're trying to estimate its value as a potential authority and as a potential hub, and so we assign it two numerical scores: $auth(p)$ and $hub(p)$. Each of these starts out with a value equal to 1, indicating that we're initially agnostic as to which is the best in either of these categories.

Now, voting – in which we use the quality of the hubs to refine our estimates for the quality of the authorities – is simply the following:

> *Authority Update Rule:* For each page p, update $auth(p)$ to be the sum of the hub scores of all pages that point to it.

On the other hand, the list-finding technique – in which we use the quality of the authorities to refine our estimates for the quality of the hubs, is the following:

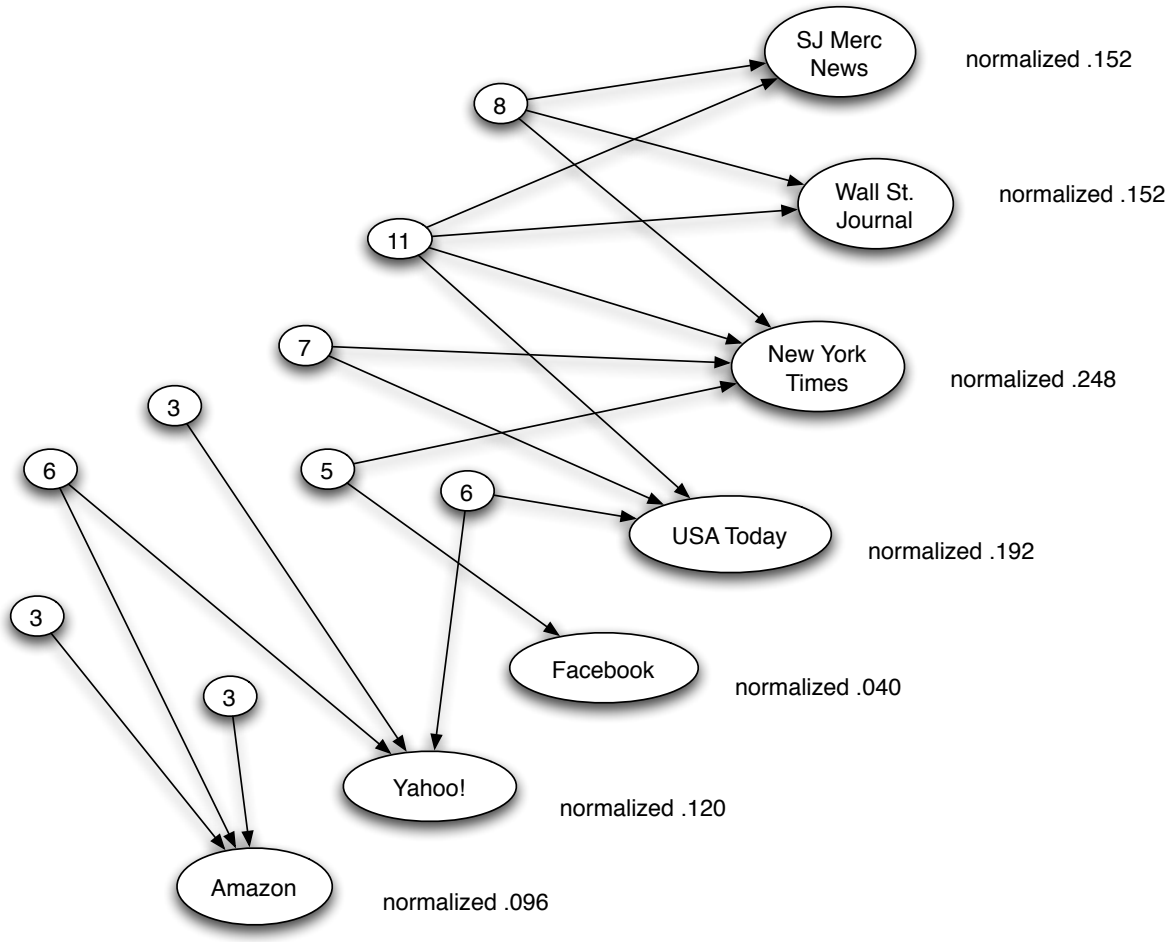

Figure 14.4: Re-weighting votes after normalizing for the query "newspapers."

Hub Update Rule: For each page p, update $hub(p)$ to be the sum of the authority scores of all pages that it points to.

Notice how a single application of the Authority Update Rule (starting from a setting in which all scores are initially 1) is simply the original casting of votes by in-links. A single application of the Authority Update Rule followed by a single application the Hub Update Rule produces the results of the original list-finding technique. In general, the Principle of Repeated Improvement says that to obtain better estimates, we should simply apply these rules in alternating fashion, as follows.

- We start with all hub scores and all authority scores equal to 1.
- We choose a number of steps k.

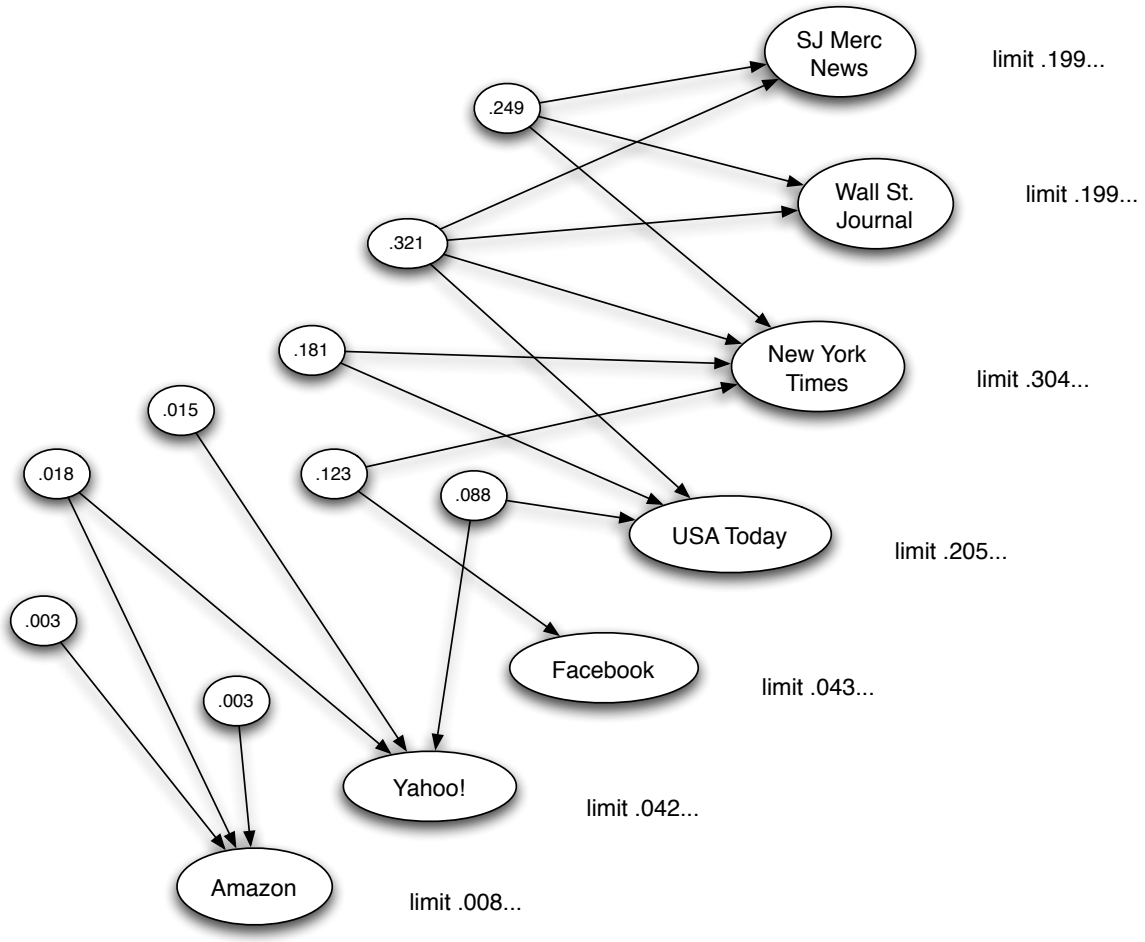

Figure 14.5: Limiting hub and authority values for the query "newspapers."

- We then perform a sequence of k hub-authority updates. Each update works as follows:
  - First apply the Authority Update Rule to the current set of scores.
  - Then apply the Hub Update Rule to the resulting set of scores.
- At the end, the hub and authority scores may involve numbers that are very large. But we only care about their relative sizes, so we can *normalize* to make them smaller: we divide down each authority score by the sum of all authority scores, and divide down each hub score by the sum of all hub scores. (For example, Figure 14.4 shows the result of normalizing the authority scores that we determined in Figure 14.3.)

What happens if we do this for larger and larger values of k? It turns out that the normalized values actually converge to limits as k goes to infinity: in other words, the

results stabilize so that continued improvement leads to smaller and smaller changes in the values we observe. We won't prove this right now, but we provide a proof in Section 14.6 at the end of this chapter. Moreover, the analysis in that section shows that something even deeper is going on: except in a few rare cases (characterized by a certain kind of degenerate property of the link structure), we reach the same limiting values no matter what we choose as the *initial* hub and authority values, provided only that all of them are positive. In other words, the limiting hub and authority values are a property purely of the link structure, not of the initial estimates we use to start the process of computing them. (For the record, the limiting values for our "newspapers" example are shown, to three decimal places, in Figure 14.5.)

Ultimately, what these limiting values correspond to is a kind of equilibrium: their relative sizes remain unchanged if we apply the Authority Update Rule or the Hub Update Rule. As such, they reflect the balance between hubs and authorities that provided the initial intuition for them: your authority score is proportional to the hub scores of the pages that point to you, and your hub score is proportional to the authority scores of the pages you point to.

14.3 PageRank

The intuition behind hubs and authorities is based on the idea that pages play multiple roles in the network, and in particular that pages can play a powerful endorsement role without themselves being heavily endorsed. For queries with a commercial aspect — such as our query for newspapers in the previous section, or searches for particular products to purchase, or more generally searches that are designed to yield corporate pages of any type — there is a natural basis for this intuition. Competing firms will not link to each other, except in unusual circumstances, and so they can't be viewed as directly endorsing each other; rather, the only way to conceptually pull them together is through a set of hub pages that link to all of them at once.

In other settings on the Web, however, endorsement is best viewed as passing directly from one prominent page to another — in other words, a page is important if it is cited by other important pages. This is often the dominant mode of endorsement, for example, among academic or governmental pages, among bloggers, or among personal pages more generally. It is also the dominant mode in the scientific literature. And it is this mode of endorsement that forms the basis for the PageRank measure of importance [79].

As with hubs and authorities, the intuition behind PageRank starts with simple voting based on in-links, and refines it using the Principle of Repeated Improvement. In particular, the Principle is applied here by having nodes repeatedly pass endorsements across their out-going links, with the weight of a node's endorsement based on the current estimate of its PageRank: nodes that are currently viewed as more important get to make stronger

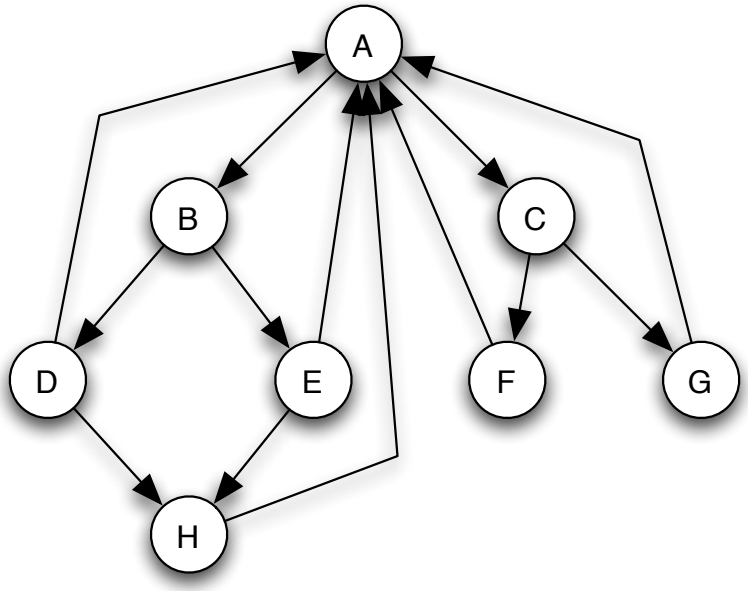

Figure 14.6: A collection of eight pages: A has the largest PageRank, followed by B and C (which collect endorsements from A).

endorsements.

The basic definition of PageRank. Intuitively, we can think of PageRank as a kind of "fluid" that circulates through the network, passing from node to node across edges, and pooling at the nodes that are the most important. Specifically, PageRank is computed as follows.

- In a network with n nodes, we assign all nodes the same initial PageRank, set to be $1/n$.
- We choose a number of steps k.
- We then perform a sequence of k *updates* to the PageRank values, using the following rule for each update:

  Basic PageRank Update Rule: Each page divides its current PageRank equally across its out-going links, and passes these equal shares to the pages it points to. (If a page has no out-going links, it passes all its current PageRank to itself.) Each page updates its new PageRank to be the sum of the shares it receives.

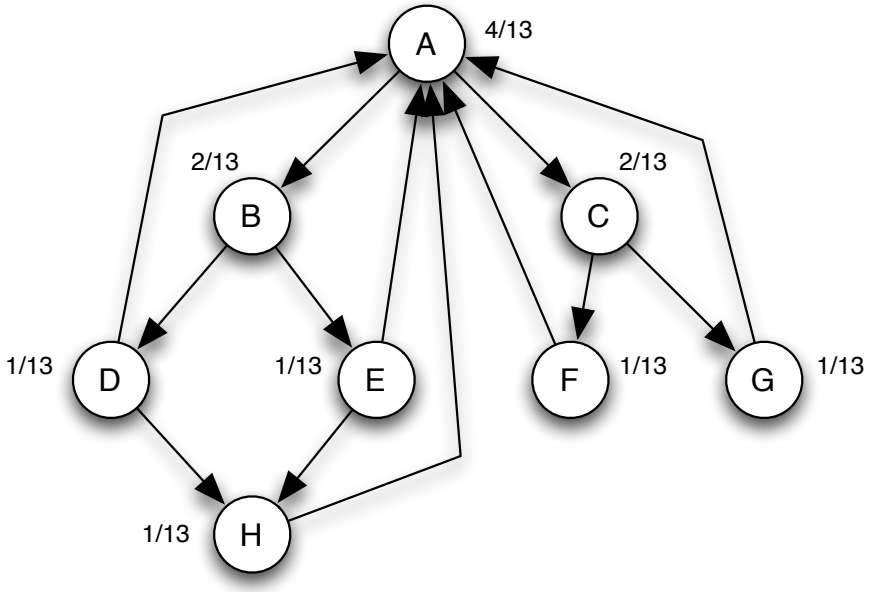

Figure 14.7: Equilibrium PageRank values for the network of eight Web pages from Figure 14.6.

Notice that the total PageRank in the network will remain constant as we apply these steps: since each page takes its PageRank, divides it up, and passes it along links, PageRank is never created nor destroyed, just moved around from one node to another. As a result, we don't need to do any normalizing of the numbers to prevent them from growing, the way we had to with hub and authority scores.

As an example, let's consider how this computation works on the collection of 8 Web pages in Figure 14.6. All pages start out with a PageRank of 1/8, and their PageRank values after the first two updates are given by the following table:

Step	A	B	C	D	E	F	G	H
1	1/2	1/16	1/16	1/16	1/16	1/16	1/16	1/8
2	3/16	1/4	1/4	1/32	1/32	1/32	1/32	1/16

For example, A gets a PageRank of 1/2 after the first update because it gets all of F's, G's, and H's PageRank, and half each of D's and E's. On the other hand, B and C each get half of A's PageRank, so they only get 1/16 each in the first step. But once A acquires a lot of PageRank, B and C benefit in the next step. This is in keeping with the principle of repeated improvement: after the first update causes us to estimate that A is an important page, we weigh its endorsement more highly in the next update.

Equilibrium Values of PageRank. As with hub-authority computations, one can prove that except in certain degenerate special cases the PageRank values of all nodes converge to limiting values as the number of update steps k goes to infinity.

Because PageRank is conserved throughout the computation — with the total PageRank in the network equal to one — the limit of the process has a simple interpretation. We can think of the limiting PageRank values, one value for each node, as exhibiting the following kind of *equilibrium*: if we take the limiting PageRank values and apply one step of the Basic PageRank Update Rule, then the values at every node remain the same. In other words, the limiting PageRank values regenerate themselves exactly when they are updated. This description gives a simple way to check whether an assignment of numbers to a set of Web pages forms such an equilibrium set of PageRank values: we check that they sum to 1, and we check that when we apply the Basic PageRank Update Rule, we get the same values back.

For example, on the network of Web pages from Figure 14.6, we can check that the values shown in Figure 14.7 have the desired equilibrium property — assigning a PageRank of $4/13$ to page A, $2/13$ to each of B and C, and $1/13$ to the five other pages achieves this equilibrium.

Now, depending on the network structure, the set of limiting values may not be the only ones that exhibit this kind of equilibrium. However, one can show that if the network is strongly connected — that is, each node can reach each other node by a directed path, following the definition from Chapter 13 — then there is a unique set of equilibrium values, and so whenever the limiting PageRank values exist, they are the only values that satisfy this equilibrium.

Scaling the definition of PageRank. There is a difficulty with the basic definition of PageRank, however: in many networks, the "wrong" nodes can end up with all the PageRank. Fortunately, there is a simple and natural way to fix this problem. yielding the actual definition of PageRank that is used in practice. Let's first describe the problem and then its solution.

To trigger the problem, suppose we take the network in Figure 14.6 and make a small change, so that F and G now point to each other rather than pointing to A. The result is shown in Figure 14.8. Clearly this ought to weaken A somewhat, but in fact a much more extreme thing happens: PageRank that flows from C to F and G can never circulate back into the rest of the network, and so the links out of C function as a kind of "slow leak" that eventually causes all the PageRank to end up at F and G. We can indeed check that by repeatedly running the Basic PageRank Update Rule, we converge to PageRank values of $1/2$ for each of F and G, and 0 for all other nodes.

This is clearly not what we wanted, but it's an inevitable consequence of the definition.

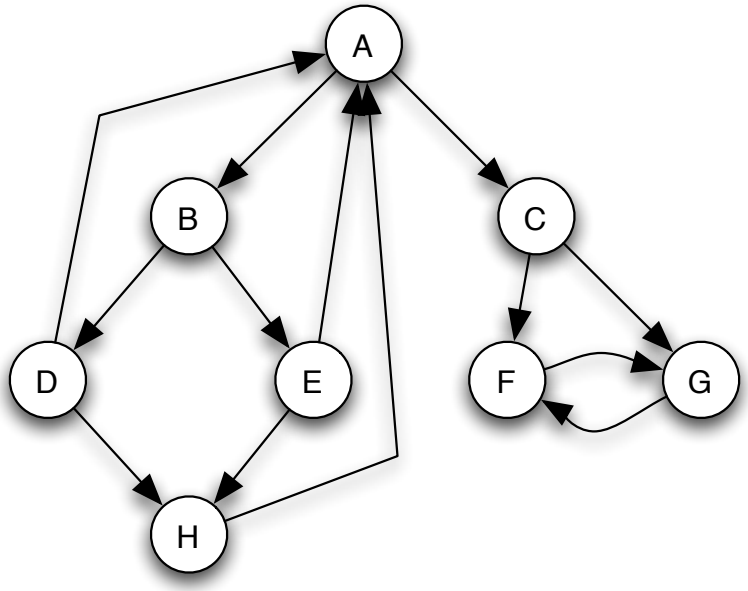

Figure 14.8: The same collection of eight pages, but F and G have changed their links to point to each other instead of to A. Without a smoothing effect, all the PageRank would go to F and G.

And it becomes a problem in almost any real network to which PageRank is applied: as long as there are small sets of nodes that can be reached from the rest of the graph, but have no paths back, then PageRank will build up there.[1] Fortunately, there is a simple and natural way to modify the definition of PageRank to get around this problem, and it follows from the "fluid" intuition for PageRank. Specifically, if we think about the (admittedly simplistic) question of why all the water on earth doesn't inexorably run downhill and reside exclusively at the lowest points, it's because there's a counter-balancing process at work: water also evaporates and gets rained back down at higher elevations.

We can use this idea here. We pick a *scaling factor* s that should be strictly between 0 and 1. We then replace the Basic PageRank Update Rule with the following:

> *Scaled PageRank Update Rule:* First apply the Basic PageRank Update Rule. Then scale down all PageRank values by a factor of s. This means that the total PageRank in the network has shrunk from 1 to s. We divide the residual $1 - s$ units of PageRank equally over all nodes, giving $(1 - s)/n$ to each.

[1] If we think back to the bow-tie structure of the Web from Chapter 13, there is a way to describe the problem in those terms as well: there are many "slow leaks" out of the giant SCC, and so in the limit, all nodes in the giant SCC will get PageRank values of 0; instead, all the PageRank will end up in the set OUT of downstream nodes.

This rule also preserves the total PageRank in the network, since it is just based on redistribution according to a different "water cycle" that evaporates $1 - s$ units of PageRank in each step and rains it down uniformly across all nodes.

The Limit of the Scaled PageRank Update Rule. One can show that repeated application of the Scaled PageRank Update Rule converges to a set of limiting PageRank values as the number of updates k goes to infinity. Moreover, for any network, these limiting values form the unique equilibrium for the Scaled PageRank Update Rule: they are the unique set of values that remain unchanged under the application of this update rule. Notice, of course, that these values depend on our choice of the scaling factor s: we should really think of there being a different update rule for each possible value of s.

This is the version of PageRank that is used in practice, with a scaling factor s that is usually chosen to be between 0.8 and 0.9.[2] The use of the scaling factor also turns out to make the PageRank measure less sensitive to the addition or deletion of small numbers of nodes or links [268, 422].

Random walks: An equivalent definition of PageRank. To conclude our discussion in this section, we now describe an equivalent formulation of PageRank that looks quite different on the surface, but in fact leads to exactly the same definition.

It works as follows. Consider someone who is randomly browsing a network of Web pages, such as the one in Figure 14.6. They start by choosing a page at random, picking each page with equal probability. They then follow links for a sequence of k steps: in each step, they pick a random out-going link from their current page, and follow it to where it leads. (If their current page has no out-going links, they just stay where they are.) Such an exploration of nodes performed by randomly following links is called a *random walk* on the network. We should stress that this is not meant to be an accurate model of an actual person exploring the Web; rather, it is a thought experiment that leads to a particular definition.

In Section 14.6, we analyze this random walk and show the following fact:

> *Claim: The probability of being at a page X after k steps of this random walk is precisely the PageRank of X after k applications of the Basic PageRank Update Rule.*

[2] As an aside about our earlier motivating example, one can check that using a value of s in this range doesn't completely fix the problem with Figure 14.8: nodes F and G still get most (though no longer all) of the PageRank under the scaled update rule with such values of s. The problem is that an eight-node example is simply too small for the redistribution of the PageRank to truly offset the problem of a slow leak into a dead-end region of the network: on only eight nodes, a "slow leak" isn't actually so slow. However, on large networks such as are used in real applications, the redistribution of PageRank works well to give nodes outside the giant strongly connected component of the network very small limiting PageRank values.

Given that the two formulations of PageRank — based on repeated improvement and random walks respectively — are equivalent, we do not strictly speaking gain anything at a formal level by having this new definition. But the analysis in terms of random walks provides some additional intuition for PageRank as a measure of importance: the PageRank of a page X is the limiting probability that a random walk across hyperlinks will end up at X, as we run the walk for larger and larger numbers of steps.

This equivalent definition using random walks also provides a new and sometimes useful perspective for thinking about some of the issues that came up earlier in the section. For example, the "leakage" of PageRank to nodes F and G in Figure 14.8 has a natural interpretation in terms of the random walk on the network: in the limit, as the walk runs for more and more steps, the probability of the walk reaching F or G is converging to 1; and once it reaches either F or G, it is stuck at these two nodes forever. Thus, the limiting probabilities of being at F and G are converging to $1/2$ each, and the limiting probabilities are converging to 0 for all other nodes.

We will also show in Section 14.6 how to formulate the Scaled PageRank Update Rule in terms of random walks. Rather than simply following a random edge in each step, the walker performs a "scaled" version of the walk as follows: With probability s, the walker follows a random edge as before; and with probability $1 - s$, the walker jumps to a random node anywhere in the network, choosing each node with equal probability.

14.4 Applying Link Analysis in Modern Web Search

The link analysis ideas described in Sections 14.2 and 14.3 have played an integral role in the ranking functions of the current generation of Web search engines, including Google, Yahoo!, Microsoft's search engine Bing, and Ask. In the late 1990s, it was possible to produce reasonable rankings using these link analysis methods almost directly on top of conventional search techniques; but with the growth and enormously expanding diversity of Web content since then, link analysis ideas have been extended and generalized considerably, so that they are now used in a wide range of different ways inside the ranking functions of modern search engines.

It is hard to say anything completely concrete about the current ranking functions of the main search engines, given that they are constantly evolving in complexity, and given that the search engine companies themselves are extremely secretive about what goes into their ranking functions. (There are good reasons for this secrecy, as we will discuss later.) But we can make general observations, coupled with sentiments that represent the conventional wisdom of the search community. In particular, PageRank was one of the original and central ingredients of Google, and it has always been a core component of its methodology. The importance of PageRank as a feature in Google's ranking function has long been claimed

to be declining over time, however. For example, in 2003 and 2004, a significant overhaul of Google's ranking function was generally believed to involve non-PageRank styles of link analysis, including a method called Hilltop [58], developed by Krishna Bharat and George Mihaila as an extension of the two-sided form of endorsement behind hubs and authorities. Around a similar time period, the search engine Ask rebuilt its ranking function around hubs and authorities, though its recent extensions have increasingly blended this in with many other features as well.

Combining links, text, and usage data. While our emphasis on link analysis in this chapter was meant to motivate the ideas in a clean setting, in practice one clearly needs to closely integrate information from both network structure and textual content in order to produce the highest-quality search results. One particularly effective way to combine text and links for ranking is through the analysis of *anchor text*, the highlighted bits of clickable text that activate a hyperlink leading to another page [102]. Anchor text can be a highly succinct and effective description of the page residing at the other end of a link; for example, if you read "I am a student at Cornell University" on someone's Web page, it's a good guess that clicking on the highlighted link associated with the text "Cornell University" will take you to a page that is in some way about Cornell.[3]

In fact, the link analysis methods we have been describing can be easily extended to incorporate textual features such as anchor text. In particular, the basic forms of both hubs and authorities and PageRank perform updates by simply adding up values across links. But if certain links have highly relevant anchor text while others don't, we can weight the contributions of the relevant links more heavily than the others; for example, as we pass hub or authority scores, or PageRank values, across a link, we can multiply them by a factor that indicates the quality of the anchor text on the link [57, 102].

In addition to text and links, search engines use many other features as well. For example, the way in which users choose to click or not click on a search result conveys a lot of information: if among a search engine's ranked results for the query "Cornell," most users skip the first result and click on the second, it suggests that the first two results should potentially be reordered. There is ongoing research on methods for tuning search results based on this type of feedback [228].

A moving target. A final important aspect of Web search serves to illustrate a basic game-theoretic principle that we have encountered many times already — that you should always expect the world to react to what you do. As search grew into the dominant means of accessing information on the Web, it mattered to a lot of people whether they ranked highly

[3] Of course, not all anchor text is useful; consider the ubiquitous bit of Web page text, "For more information, click here." Such examples make you realize that creating useful anchor text is an aspect of hypertext authoring style worth paying attention to.

in search engine results. For example, many small companies had business models that increasingly depended on showing up among the first screen of Google's results for common queries ranging from "Caribbean vacations" to "vintage records." An update to Google's ranking function that pushed them off the first screen could spell financial ruin. Indeed, search-industry publications began naming some of Google's more significant updates to its core ranking function in the alphabetic style usually reserved for hurricanes — and the analogy was an apt one, since each of these updates was an unpredictable act of nature (in this case, Google) that inflicted millions of dollars of economic damage.

With this in mind, people who depended on the success of their Web sites increasingly began modifying their Web-page authoring styles to score highly in search engine rankings. For people who had conceived of Web search as a kind of classical information retrieval application, this was something novel. Back in the 1970s and 1980s, when people designed information retrieval tools for scientific papers or newspaper articles, authors were not overtly writing their papers or abstracts with these search tools in mind.[4] From the relatively early days of the Web, however, people have written Web pages with search engines quite explicitly in mind. At first, this was often done using over-the-top tricks that aroused the ire of the search industry; as the digital librarian Cliff Lynch noted at the time, "Web search is a new kind of information retrieval application in that the documents are actively behaving badly."

Over time though, the use of focused techniques to improve a page's performance in search engine rankings became regularized and accepted, and guidelines for designing these techniques emerged; a fairly large industry known as *search engine optimization* (SEO) came into being, consisting of search experts who advise companies on how to create pages and sites that rank highly. And so to return to the game-theoretic view: the growth of SEO followed naturally once search became such a widespread application on the Web; it simply mattered too much to too many people that they be easily findable through search.

These developments have had several consequences. First, they mean that for search engines, the "perfect" ranking function will always be a moving target: if a search engine maintains the same method of ranking for too long, Web-page authors and their consultants become too effective at reverse-engineering the important features, and the search engine is in effect no longer in control of what ranks highly. Second, it means that search engines are incredibly secretive about the internals of their ranking functions — not just to prevent competing search engines from finding out what they're doing, but also to prevent designers of Web sites from finding out.

And finally, with so much money at stake, the search industry turned these developments into a very successful business model based on advertising. Rather than simply showing results computed by a ranking function, the search engine offered additional slots on the

[4] One can argue, of course, that at a less overt level, the development of standard authoring styles in these domains has been motivated by the goal of making these kinds of documents easier to classify and organize.

main results page through a market in which sites could pay for placement. Thus, when you look at a search results page today, you see the results computed by the ranking function alongside the paid results. We have just seen some of the ideas behind ranking functions; the paid results, as we will see in the next chapter, are allocated using the kinds of matching markets discussed in Chapter 10.

14.5 Applications beyond the Web

Link analysis techniques of the kind we've been discussing have been applied to a wide range of other settings, both before and after their use on the Web. In essentially any domain where information is connected by a network structure, it becomes natural to infer measures of authority from the patterns of links.

Citation Analysis. As we discussed in Chapters 2 and 13, the study of citations among scientific papers and journals has a long history that significantly predates the Web [145]. A standard measure in this field is Garfield's *impact factor* for a scientific journal [177], defined to be the average number of citations received by a paper in the given journal over the past two years. This type of voting by in-links can thus serve as a proxy for the collective attention that the scientific community pays to papers published in the journal.

In the 1970s, Pinski and Narin extended the impact factor by taking into account the idea that not all citations should be counted equally — rather, citations from journals that are themselves high-impact should be viewed as more important [341]. This can be viewed as a use of the principle of repeated improvement, in the context of the scientific literature, just as we've seen it used for Web-page ranking. Pinski and Narin used this to formulate a notion of *influence weights* for journals [180, 341] that is defined very similarly to the notion of PageRank for Web pages.

Link Analysis of U.S. Supreme Court Citations. Recently, researchers have adapted link analysis techniques from the Web to study the network of citations among legal decisions by U.S. courts [166, 377]. Citations are crucial in legal writing, to ground a decision in precedent and to explain the relation of a new decision to what has come before. Using link analysis in this context can help in identifying cases that play especially important roles in the overall citation structure.

In one example of this style of research, Fowler and Jeon applied hub and authority measures to the set of all U.S. Supreme Court decisions, a collection of documents that spans more than two centuries [166]. They found that the set of Supreme Court decisions with high authority scores in the citation network align well with the more qualitative judgments of legal experts about the Court's most important decisions. This includes some cases that

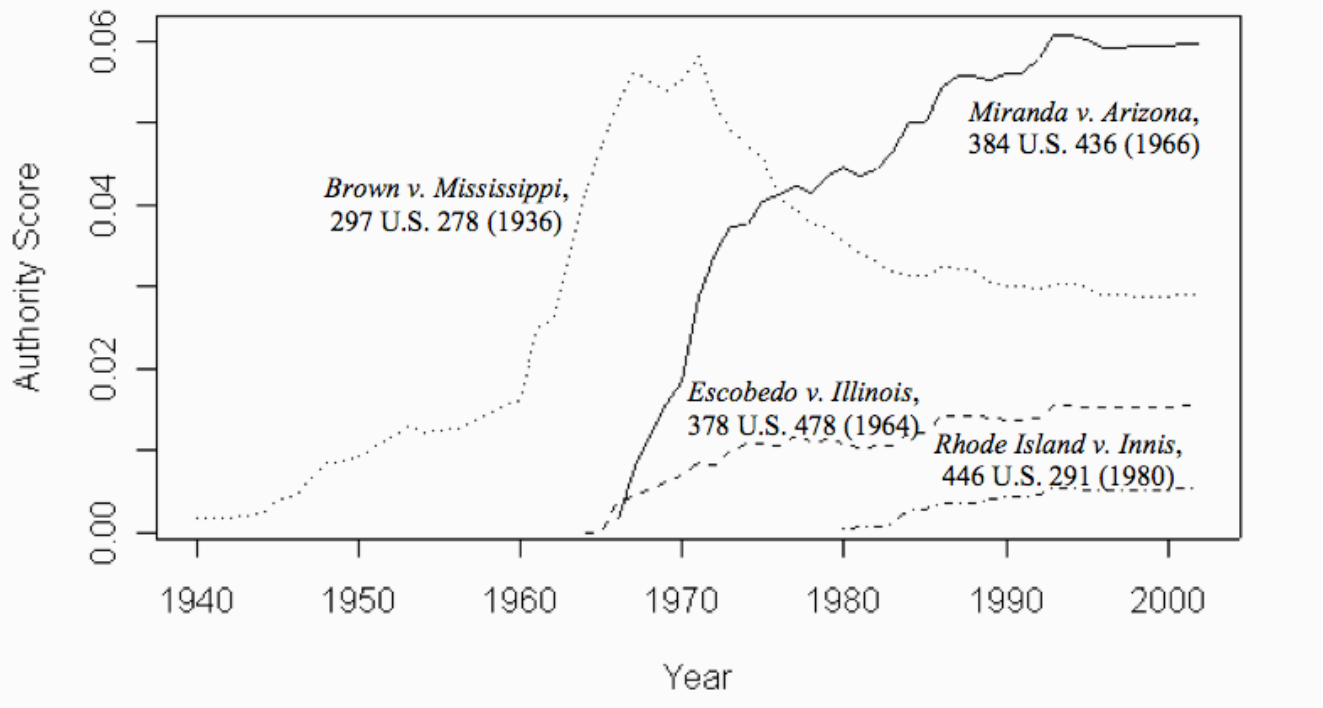

Figure 14.9: The rising and falling authority of key Fifth Amendment cases from the 20th century illustrates some of the relationships among them. (Image from [166].)

acquired significant authority according to numerical measures shortly after they appeared, but which took much longer to gain recognition from the legal community.

Supreme Court decisions also provide a rich setting for looking at how authority can change over long time periods. For example, Fowler and Jeon analyzed the rising and falling authority of some of the key Fifth Amendment cases from the 20th century, as illustrated in Figure 14.9. In particular, *Brown v. Mississippi* — a 1936 case concerning confessions obtained under torture — began rising rapidly in authority in the early 1960s as the Warren Court forcefully took on a range of issues surrounding due process and self-incrimination. This development ultimately led to the landmark case *Miranda v. Arizona* in 1966 — and with this clear precedent established, the need for citations to *Brown v. Mississippi* quickly declined as the authority of *Miranda* shot upward.

The analysis of Supreme Court citations also shows that significant decisions can vary widely in the rate at which they acquire authority. For example, Figure 14.10 (also from [166]) shows that *Roe v. Wade* — like *Miranda* — grew in authority very rapidly from the time it was first issued. On the other hand, the equally consequential *Brown v. Board of Education* only began acquiring significant authority in the citation network roughly a decade after it was issued. Fowler and Jeon argue that this trajectory aligns with legal scholars'

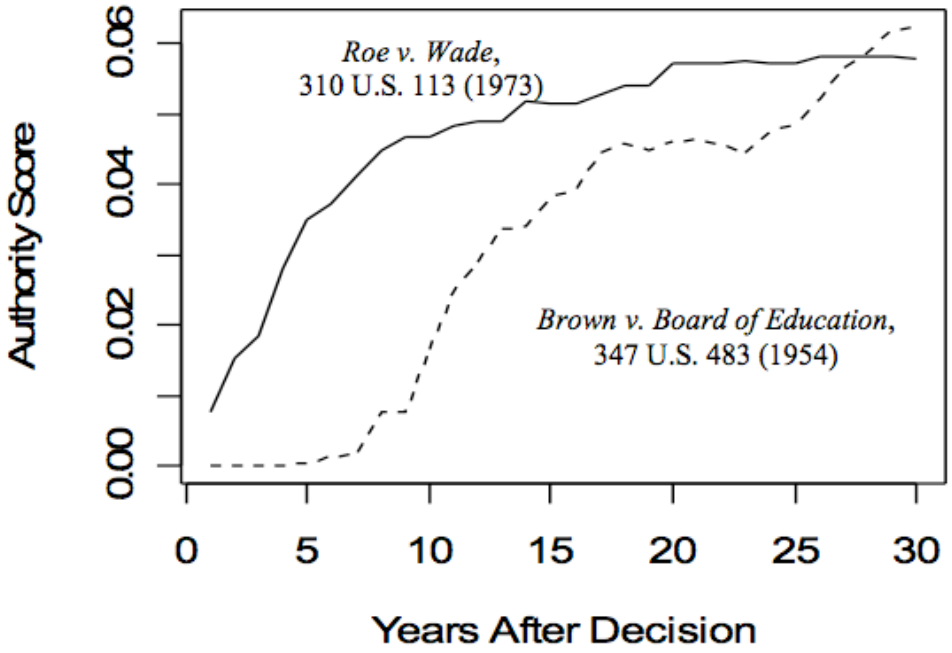

Figure 14.10: *Roe v. Wade* and *Brown v. Board of Education* acquired authority at very different speeds. (Image from [166].)

views of the case, writing, "Judicial specialists often point towards the ruling issued in *Brown* as an example of a precedent that was legally weak when first issued, and was strengthened through the Civil Rights Act of 1964 and its application in subsequent civil rights cases" [166].

This style of analysis thus shows how a strictly network-based analysis of a topic as intricate as legal precedent can reveal subtleties that align well with the views of the scholarly community. It also indicates some of the interesting effects that emerge when one tries to track the rising and falling pattern of authority in a complex domain — an activity that stands to provide important insights in many other settings as well.

14.6 Advanced Material: Spectral Analysis, Random Walks, and Web Search

We now discuss how to analyze the methods for computing hub, authority, and PageRank values. This will require some basic familiarity with matrices and vectors. Building on this, we will show that the limiting values of these link-analysis measures can be interpreted as coordinates in eigenvectors of certain matrices derived from the underlying networks. The use of eigenvalues and eigenvectors to study the structure of networks is often referred to as

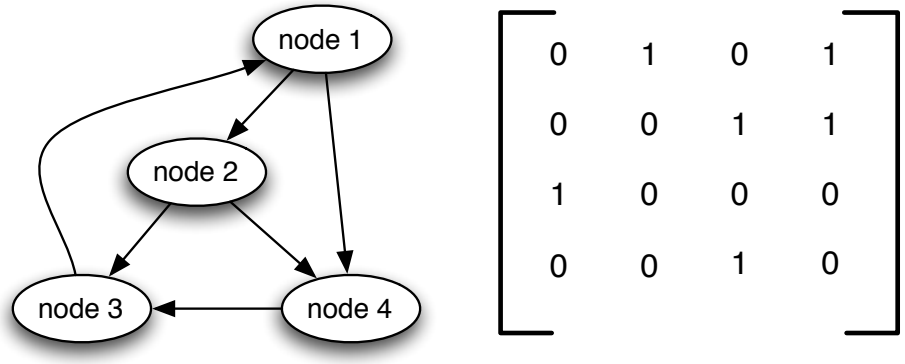

Figure 14.11: The directed hyperlinks among Web pages can be represented using an *adjacency matrix* M: the entry M_{ij} is equal to 1 if there is a link from node i to node j, and $M_{ij} = 0$ otherwise.

the *spectral analysis* of graphs, and we will see that this theory forms a natural language for discussing the outcome of methods based on repeated improvement.

A. Spectral Analysis of Hubs and Authorities

Our first main goal will be to show why the hub-authority computation converges to limiting values for the hub and authority scores, as claimed in Section 14.2. As a first important step in this, we show how to write the Authority Update and Hub Update Rules from that section as matrix-vector multiplications.

Adjacency Matrices and Hub/Authority Vectors. We will view a set of n pages as a set of nodes in a directed graph. Given this set of nodes, labeled $1, 2, 3, \ldots, n$, let's encode the links among them in an $n \times n$ matrix M as follows: the entry in the i^{th} row and j^{th} column of M, denoted M_{ij}, will be equal to 1 if there is a link from node i to node j, and it will be equal to 0 otherwise. We will call this the *adjancency matrix* of the network. Figure 14.11 shows an example of a directed graph and its adjacency matrix. Given a large set of pages, we expect that most of them will have very few outlinks relative to the total number of pages, and so this adjacency matrix will have most entries equal to 0. As a result, the adjacency matrix is not necessarily a very efficient way to represent the network, but as we will see, it is conceptually very useful.

Now, since the hub and authority scores are lists of numbers — one associated with each

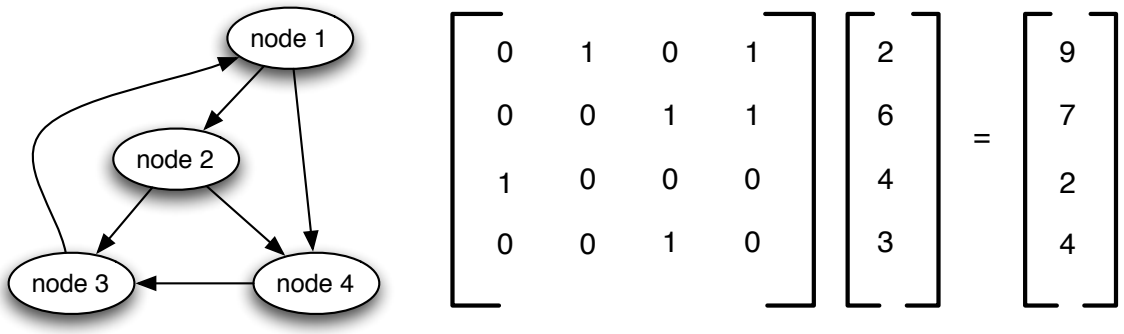

Figure 14.12: By representing the link structure using an adjacency matrix, the Hub and Authority Update Rules become matrix-vector multiplication. In this example, we show how multiplication by a vector of authority scores produces a new vector of hub scores.

of the n nodes of the network — we can represent them simply as vectors in n dimensions, where the i^{th} coordinate gives the hub or authority score of node i. Specifically, we write h for the vector of hub scores, with h_i equal to the hub score of node i, and we similarly write a for the vector of authority scores.

Hub and Authority Update Rules as Matrix-Vector Multiplication. Let's consider the Hub Update Rule in terms of the notation we've just defined. For a node i, its hub score h_i is updated to be the sum of a_j over all nodes j to which i has an edge. Note that these nodes j are precisely the ones for which $M_{ij} = 1$. Thus we can write the update rule as

$$h_i \leftarrow M_{i1}a_1 + M_{i2}a_2 + \cdots + M_{in}a_n, \tag{14.1}$$

where we use the notation "$\leftarrow$" to mean that the quantity on the left-hand-side is updated to become the quantity on the right-hand-side. This is a correct way to write the update rule, since the values M_{ij} as multipliers select out precisely the authority values that we wish to sum.

But Equation (14.1) corresponds exactly to the definition of matrix-vector multiplication, so we can write it in the following equivalent way:

$$h \leftarrow Ma.$$

Figure 14.12 shows this for the example from Figure 14.11, with the authority scores $(2, 6, 4, 3)$ producing the hub scores $(9, 7, 2, 4)$ via the Hub Update Rule. Indeed, this is an example of a general principle: if you're updating a collection of variables according to a rule that

selects out certain ones to add up, you can often write this update rule as a matrix-vector multiplication for a suitably chosen matrix and vector.

Specifying the Authority Update Rule in this style is strictly analogous, except that the scores flow in the other direction across the edges. That is, a_i is updated to be the sum of h_j over all nodes j that have an edge to i, so

$$a_i \leftarrow M_{1i}h_1 + M_{2i}h_2 + \cdots + M_{ni}h_n. \tag{14.2}$$

This too corresponds to a matrix-vector multiplication, but using a matrix where the entries have all been "reflected" so that the roles of rows and columns are interchanged. This can be specified using the *transpose* of the matrix M, denoted M^T, and defined by the property that the (i, j) entry of M^T is the (j, i) entry of M: that is, $M^T_{ij} = M_{ji}$. Then Equation (14.2) corresponds to the update rule

$$a \leftarrow M^T h.$$

Unwinding the k-step hub-authority computation. So far we have discussed a single application of each of the update rules. What happens when we perform the k-step hub-authority computation for some large value of k?

We start with initial vectors of authority and hub scores that we denote $a^{\langle 0\rangle}$ and $h^{\langle 0\rangle}$, each of them equal to the vector all of whose coordinates are 1. Now, let $a^{\langle k\rangle}$ and $h^{\langle k\rangle}$ denote the vectors of authority and hub scores after k applications of the Authority and then Hub Update Rules in order, as in Section 14.2. If we simply follow the formulas above, we first find that

$$a^{\langle 1\rangle} = M^T h^{\langle 0\rangle}$$

and

$$h^{\langle 1\rangle} = Ma^{\langle 1\rangle} = MM^T h^{\langle 0\rangle}.$$

That's the result of the 1-step hub-authority computation. In the second step, we therefore get

$$a^{\langle 2\rangle} = M^T h^{\langle 1\rangle} = M^T MM^T h^{\langle 0\rangle}$$

and

$$h^{\langle 2\rangle} = Ma^{\langle 2\rangle} = MM^T MM^T h^{\langle 0\rangle} = (MM^T)^2 h^{\langle 0\rangle}.$$

One more step makes the pattern clear:

$$a^{\langle 3\rangle} = M^T h^{\langle 2\rangle} = M^T MM^T MM^T h^{\langle 0\rangle} = (M^T M)^2 M^T h^{\langle 0\rangle}$$

and

$$h^{\langle 3\rangle} = Ma^{\langle 3\rangle} = MM^T MM^T MM^T h^{\langle 0\rangle} = (MM^T)^3 h^{\langle 0\rangle}.$$

Proceeding for larger numbers of steps, then, we find that $a^{\langle k \rangle}$ and $h^{\langle k \rangle}$ are products of the terms M and M^T in alternating order, where the expression for $a^{\langle k \rangle}$ begins with M^T and the expression for $h^{\langle k \rangle}$ begins with M. We can write this much more compactly as

$$a^{\langle k \rangle} = (M^T M)^{k-1} M^T h^{\langle 0 \rangle}$$

and

$$h^{\langle k \rangle} = (MM^T)^k h^{\langle 0 \rangle}.$$

So that's a direct picture of what's happening in the k-step hub-authority computation: the authority and hub vectors are the results of multiplying an initial vector by larger and larger powers of $M^T M$ and MM^T respectively. We now consider why this process converges to stable values.

Thinking about multiplication in terms of eigenvectors. Let's keep in mind that, since the actual magnitude of the hub and authority values tend to grow with each update, they will only converge when we take normalization into account. To put it another way, it is the *directions* of the hub and authority vectors that are converging. Concretely, what we will show is that there are constants c and d so that the sequences of vectors $\frac{h^{\langle k \rangle}}{c^k}$ and $\frac{a^{\langle k \rangle}}{d^k}$ converge to limits as k goes to infinity.

We'll talk first about the sequence of hub vectors, and then we'll consider the authority vectors largely by pursuing a direct analogy to the analysis of hub vectors. If

$$\frac{h^{\langle k \rangle}}{c^k} = \frac{(MM^T)^k h^{\langle 0 \rangle}}{c^k}$$

is going to converge to a limit $h^{\langle * \rangle}$, what properties do we expect $h^{\langle * \rangle}$ should have? Since the direction is converging, we expect that at the limit, the direction of $h^{\langle * \rangle}$ shouldn't change when it is multiplied by (MM^T), although its length might grow by a factor of c. That is, we expect that $h^{\langle * \rangle}$ will satisfy the equation

$$(MM^T) h^{\langle * \rangle} = c h^{\langle * \rangle}.$$

Any vector satisfying this property — that it doesn't change direction when multiplied by a given matrix — is called an *eigenvector* of the matrix, and the scaling constant c is called the *eigenvalue* corresponding to the eigenvector. So we expect that $h^{\langle * \rangle}$ should be an eigenvector of the matrix MM^T, with c a corresponding eigenvalue. We now prove that the sequence of vectors $\frac{h^{\langle k \rangle}}{c^k}$ indeed converges to an eigenvector of MM^T.

To prove this, we use the following basic fact about matrices. We say that a square matrix A is *symmetric* if it remains the same after transposing it: $A_{ij} = A_{ji}$ for each choice of i and j, or in other words $A = A^T$. The fact we will use is the following [268]:

Any symmetric matrix A with n rows and n columns has a set of n eigenvectors that are all unit vectors and all mutually orthogonal — that is, they form a basis *for the space $\mathbf{R}^n$.*

Since MM^T is symmetric, we can apply this fact to it. Let's write the resulting mutually orthogonal eigenvectors as $z_1, z_2, \ldots, z_n$, with corresponding eigenvalues $c_1, c_2, \ldots, c_n$ respectively; and let's order the eigenvalues so that $|c_1| \geq |c_2| \geq \cdots \geq |c_n|$. Furthermore, to make things simpler in this explanation, let's suppose that $|c_1| > |c_2|$. (This essentially always happens in link analysis applications; and below we explain the small changes that need to be made in the discussion if this assumption does not hold.) Now, given any vector x, a good way to think about the matrix-vector product $(MM^T)x$ is to first write x as a linear combination of the vectors $z_1, \ldots, z_n$. That is, with $x = p_1 z_1 + p_2 z_2 + \cdots + p_n z_n$ for coefficients $p_1, \ldots, p_n$, we have

$$\begin{aligned}(MM^T)x &= (MM^T)(p_1 z_1 + p_2 z_2 + \cdots + p_n z_n) \\ &= p_1 MM^T z_1 + p_2 MM^T z_2 + \cdots + p_n MM^T z_n \\ &= p_1 c_1 z_1 + p_2 c_2 z_2 + \cdots + p_n c_n z_n,\end{aligned}$$

where the third equality follows from the fact that each z_i is an eigenvector.

What this says is that $z_1, z_2, \ldots, z_n$ is a very useful set of coordinate axes for representing x: multiplication by MM^T consists simply of replacing each term $p_i z_i$ in the representation of x by $c_i p_i z_i$. We now see how this makes it easy to analyze multiplication by larger powers of MM^T, which will be the last step we need for showing convergence.

Convergence of the hub-authority computation. We've seen that when we take any vector x and write it in the form $p_1 z_1 + \cdots + p_n z_n$, multiplication by MM^T produces $c_1 p_1 z_1 + \cdots + c_n p_n z_n$. When we multiply repeatedly by MM^T, each successive multiplication introduces an additional factor of c_i in front of the i^{th} term. Therefore we have

$$(MM^T)^k x = c_1^k p_1 z_1 + c_2^k p_2 z_2 + \cdots + c_n^k p_n z_n.$$

Now let's think of this in the context of the vectors of hub scores, where $h^{\langle k \rangle} = (MM)^T h^{\langle 0 \rangle}$. Recall that $h^{\langle 0 \rangle}$ is just the fixed starting vector in which each coordinate is equal to 1; it can be represented in terms of the basis vectors $z_1, \ldots, z_n$ as some linear combination $h^{\langle 0 \rangle} = q_1 z_1 + q_2 z_2 \cdots + q_n z_n$. So

$$h^{\langle k \rangle} = (MM^T)^k h^{\langle 0 \rangle} = c_1^k q_1 z_1 + c_2^k q_2 z_2 + \cdots + c_n^k q_n z_n, \tag{14.3}$$

and if we divide both sides by c_1^k, then we get

$$\frac{h^{\langle k \rangle}}{c_1^k} = q_1 z_1 + \left(\frac{c_2}{c_1}\right)^k q_2 z_2 + \cdots + \left(\frac{c_n}{c_1}\right)^k q_n z_n. \tag{14.4}$$

Recalling our assumption that $|c_1| > |c_2|$ (which we'll relax shortly), we see that as k goes to infinity, every term on the right-hand side but the first is going to 0. As a result, the sequence of vectors $\frac{h^{\langle k \rangle}}{c_1^k}$ is converging to the limit $q_1 z_1$ as k goes to infinity.

Wrapping up. We're essentially done at this point; but to round out the picture of convergence, we will show two important things. First, we need to make sure that the coefficient q_1 in the argument above is not zero, so as to be able to ensure so that the limit $q_1 z_1$ is in fact a non-zero vector in the direction of z_1. Second, we will find that in fact a limit in the direction of z_1 is reached essentially *regardless* of our choice of starting hub scores $h^{\langle 0 \rangle}$: it is in this sense that the limiting hub weights are really a function of the network structure, not the starting estimates. We will show these two facts in reverse order, considering the second point first.

To begin with, then, let's suppose we began the computation of the hub vector from a different starting point: rather than having $h^{\langle 0 \rangle}$ be the vector with all coordinates equal to 1, we picked some other starting vector x. Let's suppose only that x has a positive number in each coordinate — we'll call such a vector a *positive vector*. As we noted before, any vector x can be written as $x = p_1 z_1 + \cdots p_n z_n$, for some choice of multipliers $p_1, \ldots, p_n$, and so $(MM^T)^k x = c_1^k p_1 z_1 + \cdots c_n^k p_n z_n$. Then $h^{\langle k \rangle}/c_1^k$ is converging to $p_1 z_1$ — in other words, still converging to a vector in the direction of z_1 even with this new choice for the starting vector $h^{\langle 0 \rangle} = x$.

Now, let's show why q_1 and p_1 above are not zero (hence showing that the limits are non-zero vectors). Given any vector x, there is an easy way to think about the value of p_1 in its representation as $x = p_1 z_1 + \cdots + p_n z_n$: we just compute the inner product of z_1 and x. Indeed, since the vectors $z_1, \ldots, z_n$ are all mutually orthogonal, we have

$$z_1 \cdot x = z_1 \cdot (p_1 z_1 + \cdots p_n z_n) = p_1 (z_1 \cdot z_1) + p_2 (z_1 \cdot z_2) + \cdots + p_n (z_1 \cdot z_n) = p_1,$$

since all terms in the last sum are 0 except for $p_1(z_1 \cdot z_1) = p_1$. Since p_1 is just the inner product of x and z_1, we see that our sequence of hub vectors converges to a non-zero vector in the direction of z_1 provided only that our starting hub vector $h^{\langle 0 \rangle} = x$ is not orthogonal to z_1.

We now argue that no positive vector can be orthogonal to z_1, which will conclude the picture of convergence that we've been seeking to establish. The argument works via the following steps.

1. It is not possible for every positive vector to be orthogonal to z_1, and so there is some positive vector x for which $(MM^T)^k x / c_1^k$ converges to a non-zero vector $p_1 z_1$.

2. Since the expressions for $(MM^T)^k x / c_1^k$ only involve non-negative numbers, and their values converge to $p_1 z_1$, it must be that $p_1 z_1$ has only non-negative coordinates; and $p_1 z_1$ must have at least one positive coordinate, since it is not equal to zero.

3. So if we consider the inner product of any positive vector with $p_1 z_1$, the result must be positive. Hence we conclude that *no* positive vector can be orthogonal to z_1. This establishes that in fact the sequence of hub vectors converges to a vector in the direction of z_1 when we start from *any* positive vector (including the all-ones vector), which is what we wanted to show.

This is pretty much the complete story, with the only loose end being our assumption that $|c_1| > |c_2|$. Let's now relax this assumption. In general, there may be $\ell > 1$ eigenvalues that are tied for the largest absolute value: that is, we can have $|c_1| = \cdots = |c_\ell|$, and then eigenvalues $c_{\ell+1}, \ldots, c_n$ are all smaller in absolute value. While we won't go through all the details here, it is not hard to show that all the eigenvalues of MM^T are non-negative, so in fact we have $c_1 = \cdots = c_\ell > c_{\ell+1} \geq \cdots \geq c_n \geq 0$. In this case, going back to Equations (14.3) and (14.4), we have

$$\frac{h^{\langle k \rangle}}{c_1^k} = \frac{c_1^k q_1 z_1 + \cdots + c_n^k q_n z_n}{c_1^k} = q_1 z_1 + \cdots + q_\ell z_\ell + \left(\frac{c_{\ell+1}}{c_1}\right)^k q_{\ell+1} z_{\ell+1} + \cdots + \left(\frac{c_n}{c_1}\right)^k q_n z_n.$$

Terms $\ell + 1$ through n of this sum go to zero, and so the sequence converges to $q_1 z_1 + \cdots + q_\ell z_\ell$. Thus, when $c_1 = c_2$, we still have convergence, but the limit to which the sequence converges might now depend on the choice of the initial vector $h^{\langle 0 \rangle}$ (and particularly its inner product with each of $z_1, \ldots, z_\ell$). We should emphasize, though, that in practice, with real and sufficiently large hyperlink structures, one essentially always gets a matrix M with the property that MM^T has $|c_1| > |c_2|$.

Finally, we observe that while this whole discussion has been in terms of the sequence of hub vectors, it can be adapted directly to analyze the sequence of authority vectors as well. For the authority vectors, we are looking at powers of $(M^T M)$, and so the basic result is that the vector of authority scores will converge to an eigenvector of the matrix $M^T M$ associated with its largest eigenvalue.

B. Spectral Analysis of PageRank

The analysis we've just seen emphasizes how eigenvectors arise naturally as the limits of repeated improvement. We now discuss how PageRank can be similarly analyzed using matrix-vector multiplication and eigenvectors.

Recall that like hub and authority scores, the PageRank of a node is a numerical quantity that is repeatedly refined using an update rule. Let's start by thinking about the Basic PageRank Update Rule from Section 14.3, and then move on to the scaled version. Under the basic rule, each node takes its current PageRank and divides it equally over all the nodes it points to. This suggests that the "flow" of PageRank specified by the update rule can be naturally represented using a matrix N as depicted in Figure 14.13: we define N_{ij} to be the share of i's PageRank that j should get in one update step. This means that $N_{ij} = 0$ if i

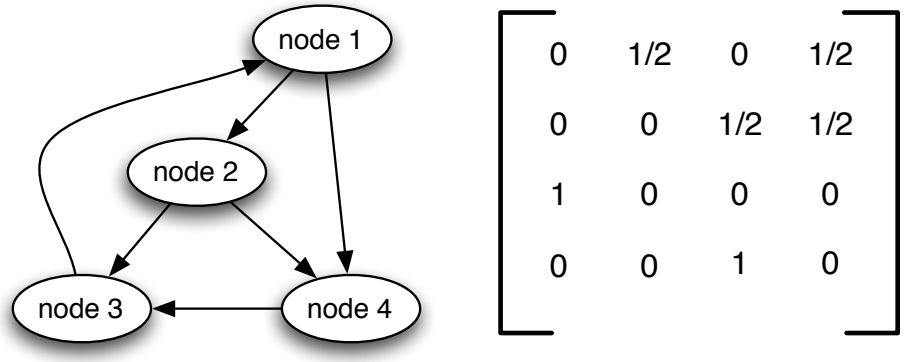

Figure 14.13: The flow of PageRank under the Basic PageRank Update Rule can be represented using a matrix N derived from the adjacency matrix M: the entry N_{ij} specifies the portion of i's PageRank that should be passed to j in one update step.

doesn't link to j, and otherwise N_{ij} is the reciprocal of the number of nodes that i points to. In other words, when i links to j, then $N_{ij} = 1/\ell_i$, where ℓ_i is the number of links out of i. (If i has no outgoing links, then we define $N_{ii} = 1$, in keeping with the rule that a node with no outgoing links passes all its PageRank to itself.) In this way, N is similar in spirit to the adjacency matrix M, but with a different definition when i links to j.

Now, let's represent the PageRanks of all nodes using a vector r, where the coordinate r_i is the PageRank of node i. Using this notation, we can write the Basic PageRank Update Rule as

$$r_i \leftarrow N_{1i}r_1 + N_{2i}r_2 + \cdots + N_{ni}r_n. \tag{14.5}$$

This corresponds to multiplication by the transpose of the matrix, just as we saw for the Authority Update Rule; thus, Equation (14.5) can be written as

$$r \leftarrow N^T r. \tag{14.6}$$

The Scaled PageRank Update Rule can be represented in essentially the same way, but with a different matrix $\tilde{N}$ to represent the different flow of PageRank, as indicated in Figure 14.14. Recall that in the scaled version of the update rule, the updated PageRank is scaled down by a factor of s, and the residual $1 - s$ units are divided equally over all nodes. Thus, we can simply define $\tilde{N}_{ij}$ to be $sN_{ij} + (1 - s)/n$, and then the scaled update rule can be written as

$$r_i \leftarrow \tilde{N}_{1i}r_1 + \tilde{N}_{2i}r_2 + \cdots + \tilde{N}_{ni}r_n. \tag{14.7}$$

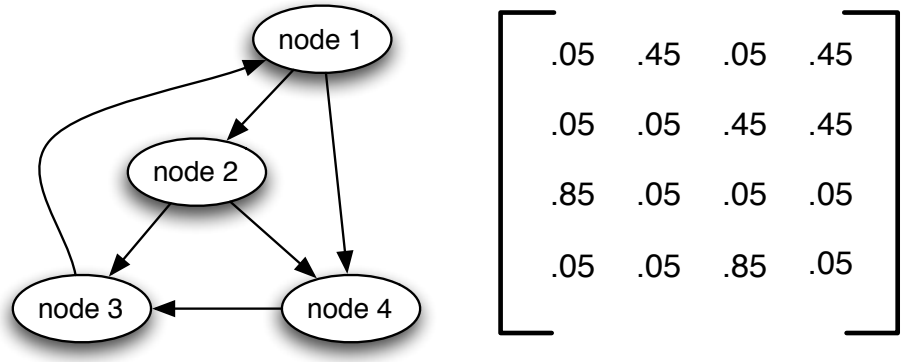

Figure 14.14: The flow of PageRank under the Scaled PageRank Update Rule can also be represented using a matrix derived from the adjacency matrix M (shown here with scaling factor $s = 0.8$). We denote this matrix by $\tilde{N}$; the entry $\tilde{N}_{ij}$ specifies the portion of i's PageRank that should be passed to j in one update step.

or equivalently

$$r \leftarrow \tilde{N}^T r. \tag{14.8}$$

Repeated Improvement Using the Scaled PageRank Update Rule. As we apply the scaled update rule repeatedly, starting from an initial PageRank vector $r^{\langle 0 \rangle}$, we produce a sequence of vectors $r^{\langle 1 \rangle}, r^{\langle 2 \rangle}, \ldots$ where each is obtained from the previous via multiplication by $\tilde{N}^T$. Thus, unwinding this process, we see that

$$r^{\langle k \rangle} = (\tilde{N}^T)^k r^{\langle 0 \rangle}.$$

Moreover, since PageRank is conserved as it is updated — that is, the sum of the PageRanks at all nodes remains constant through the application of the scaled update rule — we don't have to worry about normalizing these vectors as we proceed.

So by analogy with the limiting values of the hub-authority computation (but with the added fact that normalization isn't needed), one expects that if the Scaled PageRank Update Rule converges to a limiting vector $r^{\langle * \rangle}$, this limit should satisfy $\tilde{N}^T r^{\langle * \rangle} = r^{\langle * \rangle}$ — that is, we should expect $r^{\langle * \rangle}$ to be an eigenvector of $\tilde{N}^T$ with corresponding eigenvalue 1. Such an $r^{\langle * \rangle}$ has the property that it will not change under further refinements by the Scaled PageRank Update Rule.

In fact, all this turns out to be true: repeated application of the Scaled PageRank Update Rule converges to precisely such an $r^{\langle * \rangle}$. To prove this, however, we can't use the same

approach that we applied in the case of the hub-authority computation: there, the matrices involved (MM^T and M^TM) were symmetric, and so they had eigenvalues that were real numbers and orthogonal eigenvectors that formed a basis. In general, for matrices such as $\tilde{N}$ that are not symmetric, the eigenvalues can be complex numbers, and the eigenvectors may have less clean relationships to one another.

Convergence of the Scaled PageRank Update Rule. Fortunately, for matrices such as $\tilde{N}$ in which all entries are positive (i.e. $\tilde{N}_{ij} > 0$ for all entries $\tilde{N}_{ij}$), we can use a powerful result known as *Perron's Theorem* [268]. For our purposes, Perron's Theorem says that any matrix P in which all entries are positive has the following properties.

(i) P has a real eigenvalue $c > 0$ such that $c > |c'|$ for all other eigenvalues c'.

(ii) There is an eigenvector y with positive real coordinates corresponding to the largest eigenvalue c, and y is unique up to multiplication by a constant.

(iii) If the largest eigenvalue c is equal to 1, then for any starting vector $x \neq 0$ with non-negative coordinates, the sequence of vectors $P^k x$ converges to a vector in the direction of y as k goes to infinity.

Interpreted in terms of the (scaled) version of PageRank, Perron's Theorem tells us that there is a unique vector y that remains fixed under the application of the scaled update rule, and that repeated application of the update rule from any starting point will converge to y. This vector y thus corresponds to the limiting PageRank values we have been seeking.

C. Formulation of PageRank Using Random Walks

To close this chapter, we consider how to formulate PageRank in terms of a random walk on the nodes of the network, following the discussion at the end of Section 14.3.

First let's make the description of the random walk precise. A walker chooses a starting node at random, picking each node with equal probability. (When a random choice is made with equal probability over the options, we will say it is made *uniformly at random.*) Then, in each step, the walker follows an outgoing link selected uniformly at random from its current node, and it moves to the node that this link points to. In this way, a random path through the graph is constructed one node at a time.

Let's ask the following question: if $b_1, b_2, \ldots, b_n$ denote the probabilities of the walk being at nodes $1, 2, \ldots, n$ respectively in a given step, what is the probability it will be at node i in the next step? We can answer this by reasoning as follows.

1. For each node j that links to i, if we are given that the walk is currently at node j, then there is a $1/\ell_j$ chance that it moves from j to i in the next step, where ℓ_j is the number of links out of j.

2. The walk has to actually be at node j for this to happen, so node j contributes $b_j(1/\ell_j) = b_j/\ell_j$ to the probability of being at i in the next step.

3. Therefore, summing b_j/ℓ_j over all nodes j that link to i gives the probability the walk is at b_i in the next step.

So the overall probability that the walk is at i in the next step is the sum of b_j/ℓ_j over all nodes that link to i. We can use the matrix N defined in the analysis of PageRank to write this update to the probability b_i as follows:

$$b_i \leftarrow N_{1i}b_1 + N_{2i}b_2 + \cdots + N_{ni}b_n. \tag{14.9}$$

If we represent the probabilities of being at different nodes using a vector b, where the coordinate b_i is the probability of being at node i, then this update rule can be written using matrix-vector multiplication by analogy with what we did in our earlier analyses:

$$b \leftarrow N^T b. \tag{14.10}$$

What we discover is that this is exactly the same as the Basic PageRank Update rule from Equation (14.6). Since both PageRank values and random-walk probabilities start out the same (they are initially $1/n$ for all nodes), and they then evolve according to exactly the same rule, they remain the same forever. This justifies the claim that we made in Section 14.3:

> *Claim: The probability of being at a page X after k steps of this random walk is precisely the PageRank of X after k applications of the Basic PageRank Update Rule.*

And this makes intuitive sense. Like PageRank, the probability of being at a given node in a random walk is something that gets divided up evenly over all the outgoing links from a given node, and then passed on to the nodes at the other ends of these links. In other words, probability and PageRank both flow through the graph according to the same process.

A Scaled Version of the Random Walk. We can also formulate an interpretation of the Scaled PageRank Update Rule in terms of random walks. As suggested at the end of Section 14.3, this modified walk works as follows, for a number $s > 0$: With probability s, the walk follows a random edge as before; and with probability $1 - s$ it jumps to a node chosen uniformly at random.

Again, let's ask the following question: if $b_1, b_2, \ldots, b_n$ denote the probabilities of the walk being at nodes $1, 2, \ldots, n$ respectively in a given step, what is the probability it will be at node i in the next step? The probability of being at node i will now be the sum of sb_j/ℓ_j, over all nodes j that link to i, plus $(1 - s)/n$. If we use the matrix $\tilde{N}$ from our analysis of the Scaled PageRank Update Rule, then we can write the probability update as

$$b_i \leftarrow \tilde{N}_{1i}b_1 + \tilde{N}_{2i}b_2 + \cdots + \tilde{N}_{ni}b_n. \tag{14.11}$$

or equivalently

$$b \leftarrow \tilde{N}^T b. \tag{14.12}$$

This is the same as the update rule from Equation (14.8) for the scaled PageRank values. The random-walk probabilities and the scaled PageRank values start at the same initial values, and then evolve according to the same update, so they remain the same forever. This argument shows the following

> *Claim: The probability of being at a page X after k steps of the scaled random walk is precisely the PageRank of X after k applications of the Scaled PageRank Update Rule.*

It also establishes that as we let the number of these scaled random-walk steps go to infinity, the limiting probability of being at a node X is equal to the limiting scaled PageRank value of X.

14.7 Exercises

1. Show the values that you get if you run two rounds of computing hub and authority values on the network of Web pages in Figure 14.15. (That is, the values computed by the k-step hub-authority computation when we choose the number of steps k to be 2.)

   Show the values both before and after the final *normalization* step, in which we divide each authority score by the sum of all authority scores, and divide each hub score by the sum of all hub scores. It's fine to write the normalized scores as fractions rather than decimals.)

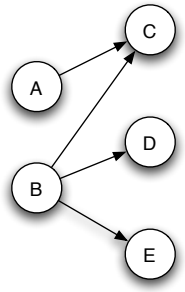

Figure 14.15:

2. **(a)** Show the values that you get if you run two rounds of computing hub and authority values on the network of Web pages in Figure 14.16. (That is, the values computed by the k-step hub-authority computation when we choose the number of steps k to be 2.)

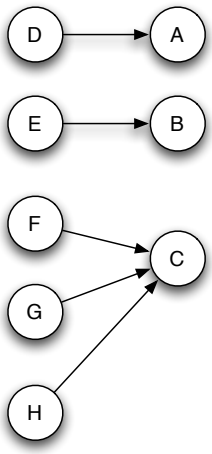

Figure 14.16: A network of Web pages.

Show the values both before and after the final *normalization* step, in which we divide each authority score by the sum of all authority scores, and divide each hub score by the sum of all hub scores. (We will call the scores obtained after this dividing-down step the *normalized scores*. It's fine to write the normalized scores as fractions rather than decimals.)

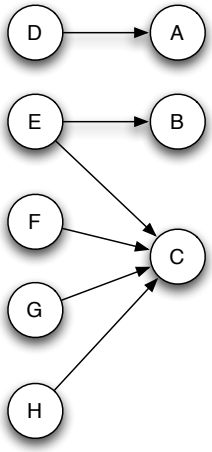

Figure 14.17: A network of Web pages.

(b) Due to the symmetry of nodes A and B in part (a), you should have seen that they get the same authority scores. Now let's look at what happens to the scores when

node E, which links to B, decides to link to C as well. This produces the new network of Web pages shown in Figure 14.17.

Similarly to part (a), show the normalized hub and authority values that each node gets when you run the 2-step hub-authority computation on the new network in Figure 14.17.

(c) In (b), which of nodes A or B now has the higher authority score? Give a brief explanation in which you provide some intuition for why the difference in authority scores between A and B in (b) turned out the way it did.

3. In Chapter 14, we discussed the fact that designers of Web content often reason explicitly about how to create pages that will score highly on search engine rankings. In a scaled-down setting, this question explores some reasoning in that style.

(a) Show the values that you get if you run two rounds of computing hub and authority values on the network of Web pages in Figure 14.18. (That is, the values computed by the k-step hub-authority computation when we choose the number of steps k to be 2.)

Show the values both before and after the final *normalization* step, in which we divide each authority score by the sum of all authority scores, and divide each hub score by the sum of all hub scores. (We will call the scores obtained after this dividing-down step the *normalized scores*. It's fine to write the normalized scores as fractions rather than decimals.)

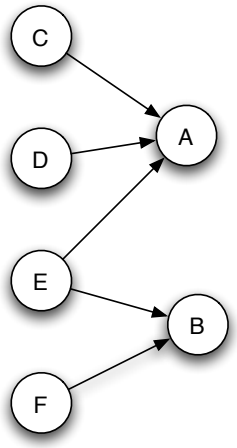

Figure 14.18:

(b) Now we come to the issue of creating pages so as to achieve large authority scores, given an existing hyperlink structure.

In particular, suppose you wanted to create a new Web page X, and add it to the network in Figure 14.18, so that it could achieve a (normalized) authority score that is as large as possible. One thing you might try is to create a second page Y as well, so that Y links to X and thus confers authority on it. In doing this, it's natural to wonder whether it helps or hurts X's authority to have Y link to other nodes as well.

Specifically, suppose you add X and Y to the network in Figure 14.18. In order to add X and Y to this network, one needs to specify what links they will have. Here are two options; in the first option, Y links only to X, while in the second option, Y links to other strong authorities in addition to X.

- *Option 1:* Add new nodes X and Y to Figure 14.18; create a single link from Y to X; create no links out of X.
- *Option 2:* Add new nodes X and Y to Figure 14.18; create links from Y to each of A, B, and X; create no links out of X.

For each of these two options, we'd like to know how X fares in terms of its authority score. So, for each option, show the normalized authority values that each of A, B, and X get when you run the 2-step hub-authority computation on the resulting network (as in part (a)). (That is, you should perform the normalization step where you divide each authority value down by the total.)

For which of Options 1 or 2 does page X get a higher authority score (taking normalization into account)? Give a brief explanation in which you provide some intuition for why this option gives X a higher score.

(c) Suppose instead of creating two pages, you create three pages X, Y, and Z, and again try to strategically create links out of them so that X gets ranked as well as possible.

Describe a strategy for adding three nodes X, Y, and Z to the network in Figure 14.18, with choices of links out of each, so that when you run the 2-step hub-authority computation (as in parts (a) and (b)), and then rank all pages by their authority score, node X shows up in second place.

(Note that there's no way to do this so that X shows up in first place, so second place is the best one can hope for using only three nodes X, Y, and Z.)

4. Let's consider the limiting values that result from the Basic PageRank Update Rule (i.e. the version where we don't introduce a scaling factor s). In Chapter 14, these limiting values are described as capturing "a kind of equilibrium based on direct endorsement: they are values that remain unchanged when everyone divides up their PageRank and passes it forward across their out-going links."

This description gives a way to check whether an assignment of numbers to a set of Web pages forms an equilibrium set of PageRank values: the numbers should add up to 1, and they should remain unchanged when we apply the Basic PageRank Update Rule. For example, this is illustrated in Chapter 14 via Figure 14.6: you can check that if we assign a PageRank of 4/13 to page A, 2/13 to each of B and C, and 1/13 to the five other pages, then these numbers add up to 1 and they remain unchanged when we apply the Basic PageRank Update Rule. Hence they form an equilibrium set of PageRank values.

For each of the following two networks, use this approach to check whether the numbers indicated in the figure form an equilibrium set of PageRank values. (In cases where the numbers do not form an equilibrium set of PageRank values, you do not need to give numbers that do; you simply need to explain why the given numbers do not.)

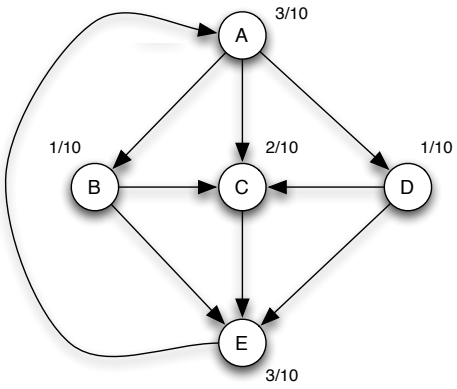

Figure 14.19: A network of Web pages.

(a) Does the assignment of numbers to the nodes in Figure 14.19 form an equilibrium set of PageRank values for this network of Web pages? Give an explanation for your answer.

(b) Does the assignment of numbers to the nodes in Figure 14.20 form an equilibrium set of PageRank values for this network of Web pages? Give an explanation for your answer.

5. Figure 14.21 depicts the links among 6 Web pages, and also a proposed PageRank value for each one, expressed as a decimal next to the node.

   Are these correct equilibrium values for the Basic PageRank Update Rule? Give a brief (1-3 sentence) explanation for your answer.

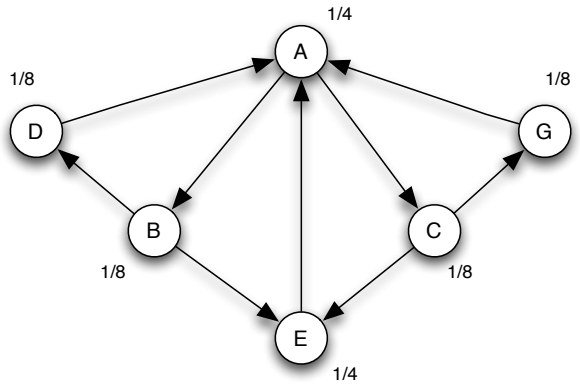

Figure 14.20: A network of Web pages.

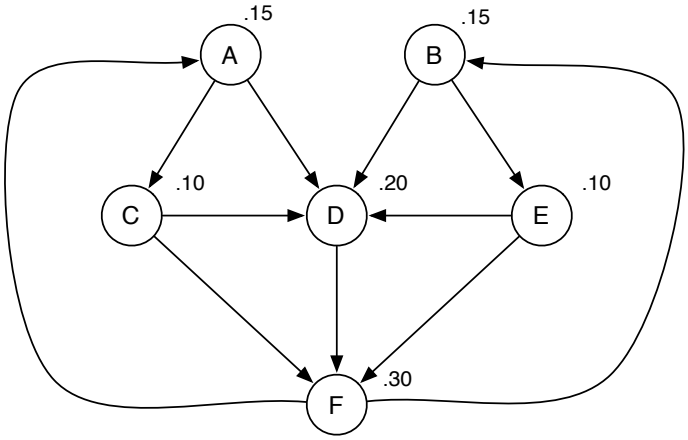

Figure 14.21: A collection of 6 Web pages, with possible PageRank values.

6. One of the basic ideas behind the computation of hubs and authorities is to distinguish between pages that have multiple reinforcing endorsements and those that simply have high in-degree. (Recall that the in-degree of a node is the number of links coming into it.)

   Consider for example the graph shown in Figure 14.22. (Despite the fact that it has two separate pieces, keep in mind that it is a single graph.) The contrast described above can be seen by comparing node D to nodes $B1$, $B2$, and $B3$: whereas D has many in-links from nodes that only point to D, nodes $B1$, $B2$, and $B3$ have fewer in-links each, but from a mutually reinforcing set of nodes.

   Let's explore how this contrast plays out in the context of this stylized example.

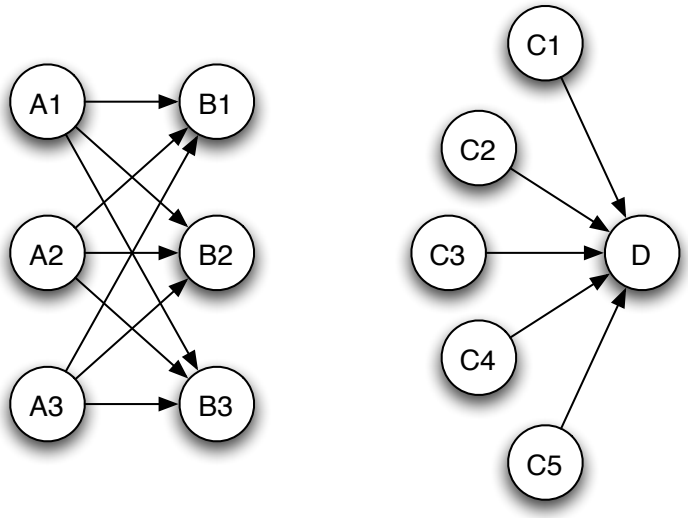

Figure 14.22:

(a) Show the values you get from running the 2-step hub-authority computation from the chapter on link analysis. (If you want, you can omit the final step in which the values are normalized; i.e., you can just leave the values as large numbers.)

(b) Give formulas, in terms of k, for the values at each node that you get from running the k-step hub-authority computation. (Again, if you want, you can omit the final step in which the values are normalized, and give the formulas in terms of k without normalization.)

(c) As k goes to infinity, what do the normalized values at each node converge to? Give an explanation for your answer; this explanation does not have to constitute a formal proof, but it should argue at least informally why the process is converging to the values you claim. In addition to your explanation of what's happening in the computation, briefly discuss (in 1-2 sentences) how this relates to the intuition suggested in the opening paragraph of this problem, about the difference between pages that have multiple reinforcing endorsements and those that simply have high in-degree.